Agent-Mediated Interface Between Command Control Communications Computer and Intelligence (C4I) Systems and Simulation

Ranjeev Mittu
Naval Research Laboratory
4555 Overlook Avenue
Washington, DC 20375
(202) 404-8716
Fax: (202) 767-1122
mittu@ait.nrl.navy.mil

Zach Furness
The MITRE Corporation
7515 Colshire Drive
McLean, VA 22102
(703) 883-6614
Fax: (703) 883-1370
zfurness@mitre.org

Ray Emami
GlobalInfoTek, Inc.
1920 Association Drive,
Reston, VA 20191
(703) 652-1600
Fax: (703) 652-1697
gemami@globalinfotek.com

Keywords: Agents, Multi-Agent Systems, Software Agents, Simulation, C4I

ABSTRACT: *The ability to interface C4I systems with simulations represents a powerful approach for analyzing complex military plans, and generating appropriate courses of action. The simulations provide a context of the real world, in which the plan can be exercised, "what-if's" can be performed, and intelligent courses of action can be generated. To build more "intelligence" in our ability to generate courses of action, we must be able to decompose plans from C4I systems, so that critical events and actions/consequence relationships can be understood. Through this understanding, we can begin to intelligently monitor how the execution of the plan may be deviating from the original simulated plan. This paper will describe technology development allowing High Level Architecture (HLA) Run Time Infrastructure (RTI)-based simulations to interact with grid-aware software agents, allowing those agents to intelligently decompose planning information from systems such the Global Command and Control System-Maritime, or GCCS-M (HLA-enabled) and monitor critical events associated with those plans within simulations. This will lead to a better understanding of the important cause-effect relationships in plans and consequently a more effective generation of courses of action.*

1. Introduction

Agent-aided information retrieval and decision support has attracted the attention of the agent research community for several years. The concept of large ensembles of semi-autonomous, intelligent agents working together is emerging as an important model for building the next generation of sophisticated software applications. This model is especially appropriate for effectively exploiting the increasing availability of diverse, heterogeneous, and distributed on-line information sources, and as a framework for building large, complex, and robust distributed information processing systems. The development of enabling infrastructure for mobile computing and interoperability among programs residing at distant sites, and new generations of distributed operating systems has made the construction of systems based on this model much easier. Software agents represent a new paradigm in distributed computing. The notion of software entities able to work autonomously, or in cooperation with each other, to perform tasks represents a powerful concept. Software agents have been deployed in many domains, ranging from the commercial, academic to the military domains.

Report Documentation Page	*Form Approved* *OMB No. 0704-0188*

1. REPORT DATE **APR 2003**	2. REPORT TYPE	3. DATES COVERED **00-00-2003 to 00-00-2003**

4. TITLE AND SUBTITLE **Agent-Mediated Interface Between Command Control Communications Computer and Intelligence (C4I) Systems and Simulation**	5a. CONTRACT NUMBER 5b. GRANT NUMBER 5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)	5d. PROJECT NUMBER 5e. TASK NUMBER 5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **Naval Research Laboratory,4555 Overlook Ave. SW,Washington,DC,20375**	8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR'S ACRONYM(S) 11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT

Approved for public release; distribution unlimited

13. SUPPLEMENTARY NOTES

Proceedings of the 2003 Spring Simulation Interoperability Workshop (SIW), Kissimmee, FL, 30 Mar ? 4 Apr 2003.

14. ABSTRACT

The ability to interface C4I systems with simulations represents a powerful approach for analyzing complex military plans, and generating appropriate courses of action. The simulations provide a context of the real world, in which the plan can be exercised, ?what-if?s? can be performed, and intelligent courses of action can be generated. To build more ?intelligence? in our ability to generate courses of action, we must be able to decompose plans from C4I systems, so that critical events and actions/consequence relationships can be understood. Through this understanding, we can begin to intelligently monitor how the execution of the plan may be deviating from the original simulated plan. This paper will describe technology development allowing High Level Architecture (HLA) Run Time Infrastructure (RTI)-based simulations to interact with grid-aware software agents, allowing those agents to intelligently decompose planning information from systems such the Global Command and Control System-Maritime, or GCCS-M (HLA-enabled) and monitor critical events associated with those plans within simulations. This will lead to a better understanding of the important cause-effect relationships in plans and consequently a more effective generation of courses of action.

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT **unclassified**	b. ABSTRACT **unclassified**	c. THIS PAGE **unclassified**	**Same as Report (SAR)**	**9**	

Over the past several years, the Defense Advanced Research Projects Agency (DARPA) has sponsored the development of the Control of Agent-Based Systems (CoABS) Grid [1]. The Grid is a middleware that enables the integration of heterogeneous agent-based systems, object-based applications and legacy systems. The CoABS grid was integrated with Critical Mission Data over Run-Time Infrastructure (RTI) (CMDR) system to allow dynamic discovery, integration and sharing of HLA compliant simulation objects with legacy C4I systems and grid-aware software agents. The development of the CMDR has paralleled, in an analogous fashion, the C4I-to-Simulation program sponsored by Defense Modeling and Simulation Organization (DMSO) that has featured the use of the High Level Architecture (HLA) RTI to pass data between systems such as the Global Command and Control System (GCCS) and the Integrated Theater Engagement Model (ITEM) [2]. The bridging of technology between the CoABS grid and the HLA using CMDR provides a wonderful opportunity to leverage the power of agent technology with the ability to tap into multiple C4I sources and simulation systems at the same time, and could lead to profound benefits in plan-understanding and execution monitoring using software agents. This paper will describe these "enabling" technologies, as well as their application for extracting C4I plan data, in order for agents to decompose and monitor that data within simulation.

This paper will begin with a description of the technology developments within the DARPA CoABS program, allowing agents to seamlessly interoperate with each other and exchange data in order to accomplish the goals of their users. Next, we will describe the developments within the C4I-to-Simulation interoperability program, specifically, the enabling technology that permits C4I systems and simulations to exchange data via the HLA RTI. We will then describe the CMDR application, allowing C4I/simulation data to be dynamically discovered and forwarded to CoABS grid agents in order for them to decompose plans and monitor crucial events within the simulations. We will conclude with a description of the planned integrated demonstration in the upcoming year, which will showcase the power of the CoABS Grid, CMDR and software agents for decomposing C4I plan data and intelligently monitoring the simulated execution of those plans.

2. The DARPA CoABS Program

The Control of Agent-Based Systems (CoABS) was a DARPA program, to develop and demonstrate techniques to safely control, coordinate, and manage large systems of autonomous software agents. CoABS was investigating the use of agent technology to improve military command, control, communication, and intelligence gathering. The military environment is dynamic, with quickly changing operations, moving hardware and software that are continually connecting and disconnecting, and bursty bandwidth availability. Inflexible stove-piped legacy systems that were never meant to be integrated are, nevertheless, of vital importance to military planning and operations. Multiple hardware and software platforms as well as data interfaces and standards further complicate the picture. In addition, military personnel are overwhelmed by the increased data availability from the modern battlefield and suffer from information overload with no adequate tools to filter and correlate the data. A goal of CoABS was to enhance the dynamic connection and operation of military planning, command, execution, and combat support systems to quickly respond to the changing operational picture. Software agents were developed to work side-by-side with human military planners and operators to ease the burden of their daily tasks.

The CoABS Grid (hereafter referred to simply as the "Grid"), developed at Global InfoTek, Inc (GITI) under the DARPA's CoABS program, arguably provides the most successful and widely used infrastructure to date for the large-scale integration of heterogeneous agent frameworks with object-based applications, and legacy systems. Based on Sun's Jini services, it includes a method-based application-programming interface to register and advertise capabilities, discover services based on those capabilities, and provides the necessary communication between services. Systems and components on the Grid can be added and upgraded without reconfiguration of the network. Failed or unavailable components are automatically purged from the registry and discovery of similar services and functionality is pursued.

The Grid supports a wide variety of applications, from simple monitoring and information retrieval to complex, dynamic domains such as military command and control. Using the Grid, agents and wrapped legacy systems can (1) describe their needs, capabilities and interfaces to other agents and legacy systems; (2) find and work with other agent components and legacy systems to accomplish complex tasks in flexible teams; (3) interact with humans and other agents to accept tasking and present results, and (4) adapt to changes in the application domain, the task at hand, or the

computing environment. The Grid does this by providing access to shared policies and ontologies (mechanisms for describing agents' capabilities and needs), and services that support interoperability among agents and legacy systems with simple or rich levels of semantics—all distributed across a network infrastructure.

Although most agent frameworks provide some of the interoperability and other services that the Grid provides, each framework typically supports specialized constructs, communication, and control mechanisms. This specialization is desirable because particular systems can use mechanisms appropriate to the problem domain/task to be solved. The Grid is not intended to replace current agent frameworks but rather to augment their capabilities with services supporting trans-architecture teams.

The Grid provides both local and distributed components, as shown in Figure 1. The Grid provides helper utility classes that are local to an agent and

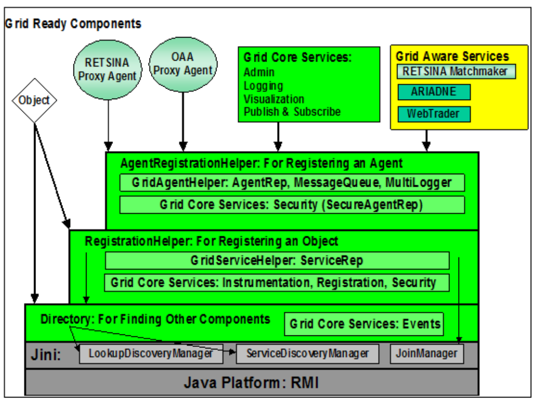

Figure 1: Grid Architecture

hide the complexity of Jini. These classes automatically find any Look-up Services (LUS) in both the local area network and user-designated distant machines. The Grid supports agent and service discovery based on Jini entries and arbitrary predicates as well as by service type. The Grid also provides event notification when agents register, deregister, or change their advertised attributes.

In the next section, we describe the developments within the C4I Simulation Interoperability Program.

3. C4I-Simulation Interface via HLA RTI

Work involving various instances of HLA-based C4I-Simulation applications has been detailed in many other papers. In 1998, DMSO sponsored an effort to utilize the HLA for passing data from the Joint Theater Level Simulation (JTLS) into the Global Command and Control System (GCCS) [3]. This application led to the development of a similar capability in which the Naval Simulation System (NSS) could stimulate GCCS, also using the HLA RTI [4]. While these applications successfully demonstrated the ability of the RTI to be used to pass information between simulations and C4I systems in much the same way that it was designed to be used between simulations, this capability was never used as part of an operational exercise.

The demonstrated utility of stimulating GCCS from JTLS and NSS using the RTI led to further applications involving the Army's Eagle simulation. In recent years, the TRADOC Analysis Center (TRAC) has sponsored development of an effort to use the RTI to pass information between Eagle and many of its C2 systems including All Source Analysis System (ASAS), Maneuver Control System (MCS), and Combat Service Support Control System (CSSCS). Details of this implementation can be found in [5].

During 2001, the GCCS-NSS capability was revived, as part of an effort to perform rapid initialization of NSS during the Global 01 exercise. Previous uses of NSS as a COAA tool in Global 00 were limited because of the need to manually input data, read off of C4I devices such as GCCS. The initialization scheme required modifications to both the GCCS RTI Interface (known as the "GCCS Ambassador") and the NSS RTI interface to allow data flow from GCCS to NSS. Details of this implementation are documented in [6]

During 2002, the GCCS-NSS initialization scheme was extended for use with the Integrated Theater Engagement Model (ITEM), an analysis application used primarily by US Pacific Command (PACOM) and United States Forces Korea (USFK). A similar scheme for initialization was developed, which relies upon data present in the GCCS Track Database Manager (TDBM) to be sent via the RTI to ITEM so that the initial state of the simulation is synchronized with GCCS as the starting point for running an analysis. This capability was successfully demonstrated in Reception Staging and Onward Integration (RSOI) 02 and Ulchi Focus Lens (UFL) 02, and will be further used during FY03 by Korea Battle Simulation Center (KBSC). Details of the implementation can be found in [7]

During 2003, DMSO is further extending the work done with NSS and ITEM to the Joint Warfare System (JWARS). An initial capability that will synchronize data from the GCCS Common Operational Picture (COP) with the current JWARS

scenario is planned to be completed by the end of 2003. This capability will help to address one of the major JWARS requirements to promote its use in Combatant Commands for in-theater analysis.

Throughout the pursuit of these efforts, a Modeling and Simulation (M&S) Technical Working Group (TWG) under the Defense Information Infrastructure Common Operating Environment (DII COE) has been working to implement the HLA RTI as a "segment" within the COE. This would allow the RTI to become part of the COE and run as a process on any command and control system that utilizes the COE. The advantage of this is that it would allow simulation applications and C4I applications to exchange data much more rapidly and efficiently, while staying within a configuration managed process (the COE) that most C4I systems utilize. Unfortunately, the history of most C4I-simulation interfaces used for training exercises is that they are implemented outside of the COE process and act as separate stand-alone processes that do not interoperate and replicate functionality. A summary of the work done to implement COE M&S segments, including the RTI can be found in [8].

In the following section, we will describe the capabilities of CMDR, and set the stage to describe the components of CMDR that will be used to act as the bridge between the C4I/Simulation worlds as well as with the software agent world in the planned integrated demonstration (section 5).

4. Current Mission Data via the RTI (CMDR)

The CMDR is a tool for developing HLA compliant applications that significantly reduces development time. CMDR has been designed and developed by GITI and is currently being used in support of a number of DARPA and DMSO sponsored initiatives. The software is a Java library designed to enable developers to quickly federate with HLA compliant simulation systems. CMDR provides a general framework for interacting with the RTI. Reusability of applications with new federations is enhanced when the applications are built using CMDR due to an independence from low-level RTI structures and data formats.

4.1 CMDR Architecture

The architecture of CMDR allows developers to rapidly develop core HLA compliant applications. The software acts as middleware between the application code and the RTI. This allows the middleware functionality to be implemented once, and can then be reused by each application through library calls. The RTI libraries and the API's provided in the HLA specification are the under-pinning of the CMDR software. Some of an application's primary responsibilities that are implemented in CMDR are:

- **Maintaining a database or internal representation of remotely simulated objects and their current states**. The RTI does not maintain a database of objects that can be queried for current attribute values by an application. It is simply the communications mechanism through which messages describing object creations, removals and attribute updates are exchanged among federates. By having this function in CMDR, an application can just query its CMDR to obtain the current state of each remote object, ignoring the details of which attributes have been updated when.

- **Managing the transmission of attribute updates for locally simulated objects**. The RTI does not keep track of the current state of the locally simulated objects either, so it can know when attribute values are out of date and thus need to be communicated to other federates.

- **Converting between raw data formats and actual objects**. The RTI transmits object attributes and interactions parameters as arrays of raw data. An important feature of CMDR is the ability to automatically translate raw data into objects and back again for many data types. This greatly reduces the amount of work necessary for examining and using the data in an application.

CMDR maintains a representation of the remote objects, adding new objects in response to the `discoverObjectInstance` RTI service, removing them in response to the `removeObjectInstance` RTI service, and updating components of their current state in response to attribute updates delivered by the `reflectAttributeValues` RTI service (attribute updates typically contain values for only a subset of an object's attributes, rather than its entire state.) Sometimes attributes are updated to their same value, for instance the heartbeat that indicates an object still exists appears as a complete update of the attributes. One of CMDR's features to improve an application's performance is the option to filter out updates that do not actually change the value. This can reduce unnecessary updates to the screen or other data models.

4.2 Agile FOM

A major issue in the development of an HLA compliant simulation is the ability of a single federate to participate in multiple federations using different Federation Object Models (FOM). Current efforts to mitigate these problems through the use of standard names and formats, while important and necessary, do not solve the problem since the ability to use different representations is a powerful feature of the HLA. Object model independence was an important consideration when developing CMDR and was the reason an internal and flexible information model was chosen. Applications built with CMDR can be quickly adapted for new federations since CMDR uses the FOM to automatically learn about the data types available and how to convert them into objects. The framework implements the agile-FOM concept by allowing the application to work with new FOMs simply by accessing those attributes that are relevant at the time. If desired, the objects and interactions found in the FOM are mapped to the applications internal object model using custom converters specified by the application.

Converters can be implemented to properly decode or encode objects moving between the internal object model and the FOM representation. When the CMDR receives an incoming attribute update, it determines the proper converter to use for decoding and converting the update. In many cases, this conversion can happen automatically based on information in the FOM. The same process occurs in reverse for outgoing updates. This allows a range of tasks, from the simple to complex, to be accomplished. For example, unit conversions between the FOM and the applications internal representation can be implemented. The ability of CMDR to use new FOMs with minimal impact allows the applications to be much more flexible and brings about additional reuse of tools between federations.

4.3 Composable Service Aware C4I Application

A C4I application has the ability to discover running agents, services, and wrapped legacy systems that are available on the network. This ability, combined with a plug-in architecture allow for vast power as the application can incorporate new capabilities by discovering and downloading remotely provided plug-ins. It puts the power of networked agents and services at the disposal of the users. As new agents or services are provided on the network, the application can instantly benefit from the new capability without having to wait for a new software rollout. During the course of an operation, if new tools are released or updates are made, the user's application discovers the available updates.

4.3.1 Plug-in Architecture

The most powerful feature of the end-user application is the architecture's ability to use software plug-ins to extend its basic capability. The architecture is designed for flexibility and reusability. The plug-ins can be provided from the local computer system or can be downloaded off the network and incorporated into the application. By using Java's introspection and reflection, the downloaded plug-in will be interrogated to determine the provided capabilities. It might be determined, for example, that the plug-in provides additional toolbar features. The plug-in architecture will add the newly discovered features to the user's C4I application toolbar.

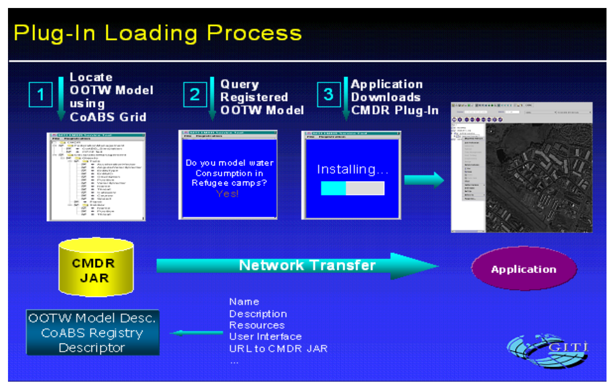

Figure 2: CMDR Plug-in Loading Process

The C4I application is not designed for use with any specific models or even for use with the HLA but through the use of a plug-in called CMDR, the architecture will incorporate the ability to become an HLA federate. The loading process is shown in Figure 2; (1) The application will query the Grid registry to locate available services. The user will see a list of the available network agents and services. (2) The user can then query the registered simulations systems to learn more about the service. In the case of a simulation model being advertised, the user might choose to investigate the purpose and assumptions of the advertised model. (3) The users application can then locate and download the necessary plug-ins to interoperate with models and simulation systems. As shown in Figure 2, a plug-in that allows the C4I application to become HLA compliant (CMDR) will be downloaded as well as a plug-in that provides additional graphical displays to view the model output.

4.3.2 Model Initialization and Tasking

A key feature of the C4I application is the capability to have the model and simulation systems initialized from remote systems and to accept tasks remotely. With these features, a remote user can initialize the model with data from a real world command and control system. This will provide the user with a model that more closely matches the factors in their situation. Tasking requests can also be made to the model to allow remote users to perform course of action analysis (COAA) and 'what-if' scenarios. All of this allows for the configuration of the underlying statistical model to test or stress the trainees decision-making process.

As a COAA tool, the model can be initialized with information from real world data. The model can then, for example, represent the population trends the logistician has been seeing over the past week. Information regarding the frequency of re-supplies can also be provided. The model would then be capable of providing feedback to the user on the projected resource situation.

4.3.3 Model Registration

A key function of the system is the ability to dynamically discover agents and services that are available on the network. To accomplish this, the simulation system needs to 'advertise' itself on the network as an available service. This advertisement will allow other agents, services, legacy systems, or applications to search for the model's offered capabilities. The advertisement consists of a description of the models capabilities, the elements of the simulation object model (SOM), and other relevant meta-data to be registered. A software plug-in is made available for download to agents, applications, or other services, which allows them to connect and interact with the model. This plug-in will allow client applications to federate with the model.

In the first step, the meta-data describing the model will be created, as shown in Figure 3. Some of this information will come directly from the software model and some from the user who is making the model available as a service. Information regarding the usage of the model, users allowed access to the model, the SOM, and other data may be provided in the meta-data advertisement. In the second step, this information is registered onto the Grid. Once in the Grid registry other services, agents, or legacy systems can dynamically discover the resource and search the meta-data to determine its appropriateness. Lastly, software plug-ins are provided for potential users of the system. The system will register, for potential users, two plug-ins. The first plug-in will provide HLA interoperability and the second will provide expanded graphical tools for C4I applications. All of this provides an advertisement for the model that allows agents and other applications to search, discover, and use the service.

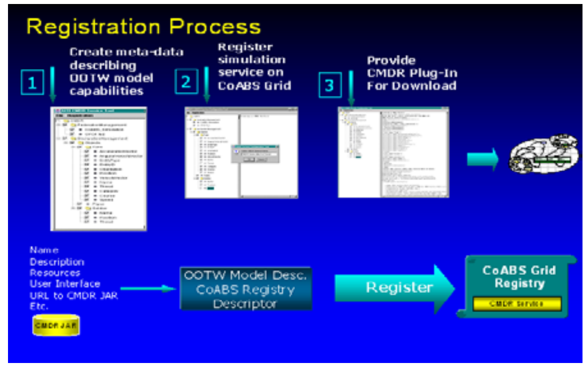

Figure 3: Model Registration Process

In the next section, we describe the integration of the RTI, CMDR and Grid to showcase the power of software agents for plan decomposition and execution monitoring.

5. Planned Integration between HLA RTI, CMDR and Grid

The planned demonstration will involve integration between the GCCS Ambassador, CMDR and the CoABS grid, in order to showcase the power of software agents for decomposing military plans, and monitoring those plans in simulation. The architecture is shown in Figure 4.

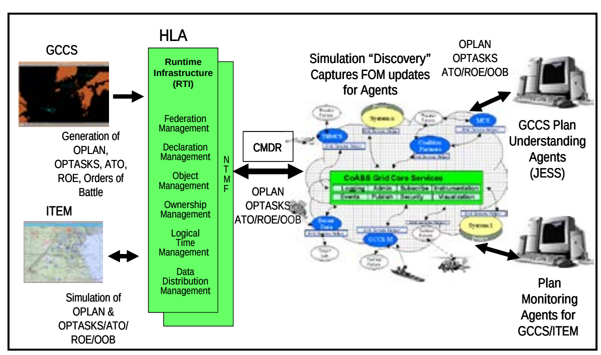

Figure 4: Integrated Demonstration Architecture

The GCCS Ambassador will publish the tracks maintained in the TDBM to the RTI (using the Naval Training MetaFOM, or NTMF [9]); once published, the ITEM simulation will use the RTI subscription mechanism to obtain those tracks. In other words, GCCS will initialize the ITEM simulation with tracks

from its database. The tracks and associated updates will be passed from CMDR to the CoABS Grid.

Plan-understanding agents registered on the Grid will be capable of decomposing external planning information, for example, from Operational Tasking (OPTASK) messages, or Air Tasking Orders (ATO's) and forming relationships between events both spatially and temporally, as well as linking those relationships with the track information from the RTI. Having formed these relationships and communicating these to the monitoring agents, the latter will begin to monitor both ITEM (which represents how entities should be moving and interacting) with GCCS (representing how entities are actually moving and interacting based on a re-play of a scenario) in order to take note of the differences that are occurring with regard to critical movements and relationships. In other words, the agents will monitor only those events and relationships that are deemed of critical importance with regard to some measurement criteria. By measuring deviations in those critical events such as rendezvous points, temporal delays in events that may impact future events, (and not every deviation occurring in the simulation, since a local deviation does not necessarily imply that a mission will not succeed), the user will be better able to comprehend the simulation. This will allow the user to perform better courses of action since now that user has a better understanding of the relationships between key events, and the agents can perform notification based on those dependencies as the courses of action are performed particularly when the constraints imposed by those dependencies are violated. In a large and complex scenario, visual detection of such dependencies will be difficult, and automation through software agents will be valuable.

5.1 Multi-Agent System (MAS) Infrastructure

This integration of the components contained within this demonstration architecture will require, among other things, additional research with regard to the development of the software agent infrastructure. One of the key areas of investigation will be in the area of agent ontology. An ontology is used to describe the objects or entities, their relationships with each other and other objects, their attributes, etc so that agents have an understanding of their world and are able to reason intelligently about their world. An ontology is also important when you consider the ability to reason about relationships in events, particularly when those events are related spatially and temporally. With regard to temporality, for example, and agent must be able to understand the concepts associated with time such as time instants, time intervals and durations, etc. The relationships associated with time may be represented in an ontology that defines these concepts. With regard to events that are spatially represented, the ontology will define what those events are and where they occur. An agent needs to understand an event ontology to be able to reason about events, particularly if there is a need to reason about things such as two or more events occurring in close proximity (like aircraft refueling point). The reason an agent needs to understand *both* a temporal and spatial ontology is that, using the example of aircraft refueling point, it must be able to reason about this refueling not only occurring at some point in space, but also at some specified time and for some duration. The DARPA Agent Markup Language (DAML) [10] is a candidate technology that may serve the purpose of building the needed ontological references for the software agents. The DAML language is an extension of the eXtensible Markup Language (XML) [11] and the Resource Description Framework (RDF) [11]. There are several efforts ongoing within the DAML community that may be leveraged, for example, the time ontology effort. Other possible candidates include XML schema and RDF schema.

There are several advantages to utilizing software agents (vice a federate that performs the plan decomposition and monitoring functions) in this architecture. One of the biggest advantages lies in the ability of software agents to understand an ontological description (or inter-related ontological descriptions) in order to reason about their world. One may argue that a federate could perform the same functions, as one might claim that a FOM also loosely resembles an ontology. However, the field of Artificial Intelligence, from which software agents have emerged, provides a richer set of technologies for software agents. For example, research in this field is examining ontology negotiation techniques to allow software agents to negotiate between the meanings of their respective ontology, thereby permitting agents to "on-the-fly" understand and reason about these new concepts based on how it relates to their own internal knowledge. In our example of aircraft refueling, perhaps additional agents with a similar ontology to our event ontology could be discovered on the agent grid to provide additional critical information about the refueling event as well that may not have been a part of the original event ontology. In comparison, the HLA does not permit a new federate to join a federation unless it uses a pre-specified FOM (i.e., a federate that may provide useful information, but uses a

different FOM, would not be able to communicate and exchange meaningful data with the federation unless it used the same FOM as the federation it is joining or is bridged through a third federate). Even the use of converters, as discussed previously in the paper, relies on a-priori development and implementation as opposed to a more dynamic approach to understanding an ontological description at run-time (although agents capable of negotiating to understand the ontological descriptions of plug-in's may provide a solution). Research in ontology negotiation techniques is still in the early stages, but there are promising approaches [12]

The second advantage is in the ability of agent's to understand multiple, unique ontological descriptions. This approach provides a more flexible distributed computing environment. In comparison, if one were to encode all of the knowledge in a FOM, then, very quickly, the FOM could become large and difficult to use and eventually maintain. Having agents understand multiple ontological descriptions and negotiate (as was just previously discussed) on those terms that are unfamiliar, can provide a much robust distributed computing solution.

The third primary benefit of the use of software agents comes from research being done in field of agent teamwork theory and models. There has been significant research in this area examining how teams of agents cooperate with each other to form beliefs about the world, and how and when they take action in order to reach a team goal. One could imagine within this architecture how teams of agents with varying capabilities are able to decompose various aspects of the plans, monitor those plans, and perhaps even aid in repairing the plans based on the outcome of the simulated results. These teams of agents may converse with other teams of agents (or even teams of users) with differing ontological knowledge and negotiate meanings of information in reaching their goals in the process of performing courses of action analysis. As with ontology negotiation techniques, there are many research topics to be addressed in making this a reality, but the basic research is being conducted in this area and is being pushed to solve practical problems.

6. Conclusion and Future Direction

This paper has presented research that is being conducted in order to bring together HLA-compliant simulations with multi-agents systems. We have described enabling technology that provides the bridge between these two "worlds" and the utility of using agents for monitoring plan data within simulation in order to conduct more effective courses of action. We have described an initial architecture, but there are many opportunities to build upon this initial research. One specific area we would like to investigate is the integration of our architecture with a formal approach to representing plans and their interdependencies. This will allow agents to interrogate the output from these systems to get a better handle of the relationships. An example system might be the Interactive Decision Support (IDS) that uses a Microsoft project interface to represent such dependencies.

Additional topics for investigation include federating additional simulations that provide specific and unique capabilities, integrating additional agent-based products emanating from the CoABS program, and expanding the capabilities of the plan-understanding and monitoring agents.

A candidate simulation of interest is the Network Warfare Simulation (NETWARS) [13]. NETWARS is an HLA compliant simulation that provides the capability to analyze communications effects on the battlefield. An effort to link NETWARS and ITEM for purposes of conducting synchronous planning is scheduled to be sponsored by DMSO during FY03. This will allow the effects of the communication infrastructure to be taken into consideration during development and refinement of an OPLAN that is being generated using ITEM.

With regard to integrating with CoABS products and research, the area dealing with agent teamwork theories and models are of particular interest in order to support the capability of teams of agents (perhaps with different ontological representations) working together to decompose and monitor plans, and propose COA solutions based on individual and team goals.

7. References

[1] Web Address: http://coabs.globalinfotek.com

[2] Furness et al. "Real-time Initialization of Planning and Analysis Simulations based on C4ISR System Data". In Proceedings of the 2002 Command and Control Research and Technology Symposium (CCRTS), Monterey, CA 11-13 June 2002.

[3] Nielsen, J., Salisbury, M., "Challenges in Developing the JTLS-GCCS-NC3A Federation", in proceedings of the 1998 Fall Simulation Interoperability Workshop (SIW), Orlando, Florida, 14-18 September 1998.

[4] Lutz, R., Salisbury, M., Bidwell, G., "A Demonstration of C2I System-to-Simulation Interoperability: The NSS/GCCS-M Federation", in proceedings of the 1999 Fall Simulation Interoperability Workshop (SIW), Orlando, Florida, 12-17 September 1999.

[5] Ogren, J., "Command and Staff Training and the Practical use of the HLA", in proceedings of the 2000 Fall Simulation Interoperability Workshop (SIW), Orlando, Florida, 17-22 September 2000.

[6] Prochnow, D., King, R., Harrington, J., Regala, B., Sonnenshein, J., Daly, J., Womble, J., "The NSS-GCCS Federation: Course of Action Analysis (COAA) Using a C4I-Simulation Interface", in proceedings of the 2001 Fall Simulation Interoperability Workshop (SIW), Orlando, Florida, 10-14 September 2001.

[7] Prochnow, D., Harrington, J., Walter, D., Womble, J., "Automate Initialization of an Analysis Simulation With GCCS Track Data", in proceedings of the 2002 Fall Simulation Interoperability Workshop (SIW), Orlando, Florida, 9-13 September 2002.

[8] Layman, G., Daly, J., Furness, Z., Womble, J., "C4I-Simulation Interoperability Using the HLA and DII COE", in proceedings of the 2001 Command Control Research and Technology Symposium (CCTRS), Annapolis, MD, 19-21 June 2001.

[9] Mittu et al., "A Case Study for the Naval Training Meta FOM: Analyzing the Requirements from MAGTF FOM". 7th International Command and Control Research and Technology Symposium, Quebec City, Canada. 9-12 Sept. 2002.

[10] Web Address: http://www.daml.org

[11] Web Address: http://www.w3c.org

[12] Tamma et al, "An Ontology for Automated Negotiation". In Proceedings of the International Workshop on Ontologies in Agent Systems. AAMAS 02 Conferece, Bologna, Italy

[13] Web Address: http://www.disa.mil/tis/netwars.

8. Acknowledgements

The authors would like to thank the Defense Modeling and Simulation Office (DMSO) for providing the research funds for this effort.

Author Biographies

RANJEEV MITTU is a Program Manager at the Naval Research Laboratory in Washington, DC. Mr. Mittu has ten years experience in designing and developing decision support systems for the Naval community. His experiences include managing programs ranging from the Strategic Defense Initiative (SDI), Anti-Submarine Warfare (ASW) as well as Force Level Planning for the Naval Strike community. He has a Master of Science degree in Electrical Engineering from The Johns Hopkins University in Baltimore, MD.

ZACH FURNESS is an Associate Department Head at the MITRE Corporation and technical lead supporting the C4I-Simulation Interoperability program at the Defense Modeling and Simulation Office (DMSO). Since 2000, he has overseen the implementation of automated initialization of simulations directly from C4I systems such as GCCS and worked to integrate simulation capabilities into the DII COE. Prior to supporting DMSO, Mr. Furness led the MITRE Systems Engineering team that supported the development, integration, and testing of the Joint Training Confederation (JTC). Since joining MITRE in 1990, Mr. Furness has worked on a variety of M&S projects related to the application of simulation in the training, analysis, and acquisition of C4ISR systems and has published numerous papers on the subject. He holds an MS in Electrical Engineering and BS in Physics, both from Virginia Tech.

DR. RAY EMAMI is the CEO of Global InfoTek, Inc. (GITI) which he founded in 1996. GITI is a leader in cutting-edge software development and integration. Dr. Emami is a technical visionary, who has over 25 years experience conceiving, creating, and leading cutting-edge technology development projects. Under his leadership, GITI has become a premier contractor supporting organizations such as the Defense Advanced Research Project Agency (DARPA), and organizations in the Intelligence Community. Under his direction, GITI has developed an impeccable reputation for creating novel technologies and transitioning advanced DARPA technologies to the warfighter.